Dynamical Numerics for Numerical Dynamics

Robert H. C. Moir

The University of Western Ontario
Dept. of Applied Mathematics

April 9, 2010

- Suppose that we want to model the time evolution of some natural or social system.

- Suppose that we want to model the time evolution of some natural or social system.
- Often this is done by devising a rule according to which the state of the system at a time t depends on the state of the system at an earlier time.

- Suppose that we want to model the time evolution of some natural or social system.
- Often this is done by devising a rule according to which the state of the system at a time t depends on the state of the system at an earlier time.

$$\text{finite earlier time} \longrightarrow \text{difference equations}$$

- Suppose that we want to model the time evolution of some natural or social system.
- Often this is done by devising a rule according to which the state of the system at a time t depends on the state of the system at an earlier time.

finite earlier time $\longrightarrow$ difference equations

discrete dynamical system:

$$U_{n+1} = f(U_n), \quad U_0 = U, \quad n \in \mathbb{N}, \mathbb{Z}, u \in \mathbb{R}^n$$

- Suppose that we want to model the time evolution of some natural or social system.
- Often this is done by devising a rule according to which the state of the system at a time t depends on the state of the system at an earlier time.

$$\text{infinitesimal earlier time} \longrightarrow \text{differential equations}$$

- Suppose that we want to model the time evolution of some natural or social system.
- Often this is done by devising a rule according to which the state of the system at a time t depends on the state of the system at an earlier time.

infinitesimal earlier time $\longrightarrow$ differential equations

continuous dynamical system:

$$\dot{u}(t) = f(u(t)), \quad u(0) = u_0, \quad n \in \mathbb{R}^+, \mathbb{R}, u \in \mathbb{R}^n \qquad (1)$$

- If f is a linear function of u, the system can be solved exactly.

- If f is a linear function of u, the system can be solved exactly.
- In most modeling situations, however, f is nonlinear and in general analytic solutions are not available.

So what can we do?

So what can we do?

1. Use qualitative analytic methods (*e.g.* centre manifold theory, normal form theory)

So what can we do?

1. Use qualitative analytic methods (*e.g.* centre manifold theory, normal form theory)
2. Use computers and numerical methods

So what can we do?

1. Use qualitative analytic methods (*e.g.* centre manifold theory, normal form theory)
2. Use computers and numerical methods

Both approaches have limitations...

Two important limitations of using computation:

Two important limitations of using computation:

1. computer representation of real numbers (floating point numbers) is finite and discrete

Two important limitations of using computation:

1. computer representation of real numbers (floating point numbers) is finite and discrete
    - this introduces roundoff and truncation error

Two important limitations of using computation:

1. computer representation of real numbers (floating point numbers) is finite and discrete
   - this introduces roundoff and truncation error
2. the continuous evolution of (1) is discretized

Two important limitations of using computation:

1. computer representation of real numbers (floating point numbers) is finite and discrete
   - this introduces roundoff and truncation error
2. the continuous evolution of (1) is discretized

$$u(t + \Delta t) = u(t) + \int_t^{t+\Delta t} f(u)dt$$

Two important limitations of using computation:

1. computer representation of real numbers (floating point numbers) is finite and discrete
   - this introduces roundoff and truncation error
2. the continuous evolution of (1) is discretized

$$u(t+\Delta t) = u(t) + \int_t^{t+\Delta t} f(u)dt \xrightarrow{\text{e.g.}} U_{n+1} = U_n + \sum_{i=1}^{s} a_i f(Y_i)\Delta t$$

Two important limitations of using computation:

1. computer representation of real numbers (floating point numbers) is finite and discrete
   - this introduces roundoff and truncation error
2. the continuous evolution of (1) is discretized
   - this introduces *discretization error*

$$u(t+\Delta t) = u(t) + \int_t^{t+\Delta t} f(u)dt \xrightarrow{\text{e.g.}} U_{n+1} = U_n + \sum_{i=1}^{s} a_i f(Y_i)\Delta t$$

So, not only do we have the issue of floating point error, we also have the issue of whether interpolating the discretized system produces behaviour sufficiently close to the behaviour of the model

Consider a very simple numerical method (Euler method):

Consider a very simple numerical method (Euler method):

- Suppose our model is the logistic equation

$$\dot{u} = f(u) = u - u^2, \quad u(0) = u_0,$$

Consider a very simple numerical method (Euler method):

- Suppose our model is the logistic equation

$$\dot{u} = f(u) = u - u^2, \quad u(0) = u_0,$$

- which has solution

$$u(t) = u_0 + \int_0^t f(u)dt = u_0 + \int_0^t (u - u^2)dt.$$

Consider a very simple numerical method (Euler method):

- Suppose our model is the logistic equation

$$\dot{u} = f(u) = u - u^2, \quad u(0) = u_0,$$

- which has solution

$$u(t) = u_0 + \int_0^t f(u)dt = u_0 + \int_0^t (u - u^2)dt.$$

- Then with the very simple estimation of the integral over a time step Δt as $f(u(t))\Delta t$ we obtain

$$U_{n+1} = U_n + (U_n - U_n^2)\Delta t,$$

Consider a very simple numerical method (Euler method):

- Suppose our model is the logistic equation

$$\dot{u} = f(u) = u - u^2, \quad u(0) = u_0,$$

- which has solution

$$u(t) = u_0 + \int_0^t f(u)dt = u_0 + \int_0^t (u - u^2)dt.$$

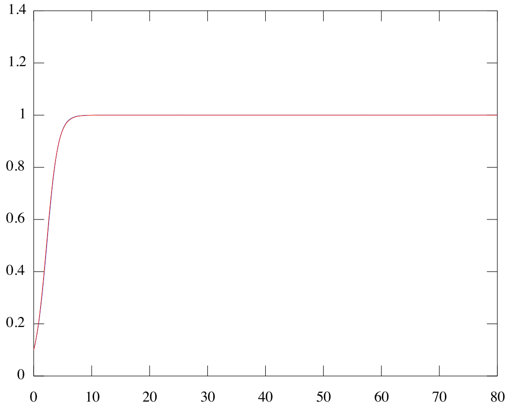
- Then with the very simple estimation of the integral over a time step Δt as $f(u(t))\Delta t$ we obtain

$$U_{n+1} = U_n + (U_n - U_n^2)\Delta t,$$

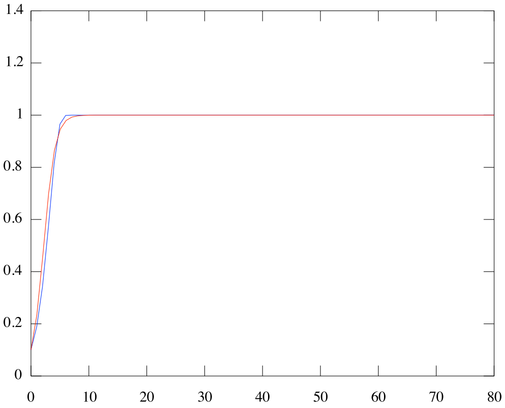
- which contains the logistic map.

- How does the Euler method do for $\Delta t = 0.1$?

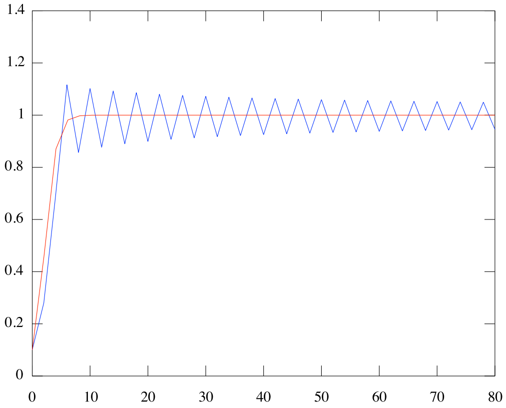
- How does the Euler method do for $\Delta t = 0.1$?



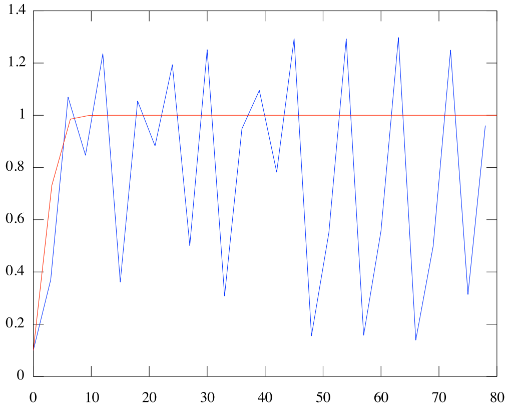
- How does the Euler method do for $\Delta t = 1$?



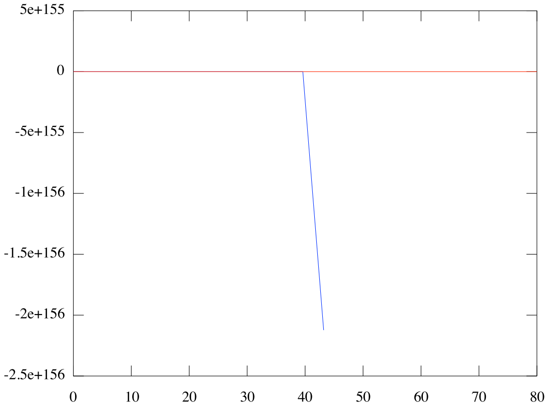
- How does the Euler method do for $\Delta t = 2$?



- How does the Euler method do for $\Delta t = 3$?



- How does the Euler method do for $\Delta t = 3.6$?



Of course this example is artificial for two reasons:

Of course this example is artificial for two reasons:

1. In practice we use more sophisticated numerical methods;

Of course this example is artificial for two reasons:

1. In practice we use more sophisticated numerical methods; and
2. In practice we use very small time steps for accuracy.

Of course this example is artificial for two reasons:

1. In practice we use more sophisticated numerical methods; and
2. In practice we use very small time steps for accuracy.

But, it shows that even though the solution of the model may be smooth, the solution of the numerics need not be.

Of course this example is artificial for two reasons:

1. In practice we use more sophisticated numerical methods; and
2. In practice we use very small time steps for accuracy.

But, it shows that even though the solution of the model may be smooth, the solution of the numerics need not be. The numerics can even be chaotic!

Since we want the numerics to accurately reflect the dynamics, so that we can use the numerics to understand the system being modeled, we must keep a careful watch on the various sources of error.

- In many modeling situations, ensuring that

$$\|u(t) - U(t)\|, \quad t \in [0, T]$$

is small is one's main concern, since one requires accurate quantitative results in the near term (short time periods).

- In many modeling situations, ensuring that

$$\|u(t) - U(t)\|, \quad t \in [0, T]$$

is small is one's main concern, since one requires accurate quantitative results in the near term (short time periods).

- The classical error theory for numerical methods formulates this as a *convergence* question.

- The error committed over one iteration of the numerical method starting at $U_0 = U$ is called the *truncation error*, which may be denoted

$$\|T(U; \Delta t)\|.$$

- The error committed over one iteration of the numerical method starting at $U_0 = U$ is called the *truncation error*, which may be denoted

$$\|T(U;\Delta t)\|.$$

- For example, for Runge-Kutta methods

$$\|T(U;\Delta t)\| = u(\Delta t) - U_1.$$

The questions asked, then, are:

The questions asked, then, are:

1. does $\|T(U;\Delta t)\| \to 0$ as $\Delta t \to 0$?

The questions asked, then, are:

1. does $\|T(U;\Delta t)\| \to 0$ as $\Delta t \to 0$?
2. and if so, how fast?

To phrase the sort of result that is proved, we need the notion of the *order* of a numerical method:

To phrase the sort of result that is proved, we need the notion of the *order* of a numerical method:

- Roughly speaking, a numerical method has *order* r if for all sufficiently smooth functions $f(u)$ and all initial values U

$$\|T(U;\Delta t)\| = O(\Delta t^{r+1}) \text{ as } \Delta t \to 0.$$

- Defining the *global error* e_n at time $t = n\Delta t$ to be

$$e_n = \|u(n\Delta t) - U_n\|$$

then an example of a convergence result is (for Runge-Kutta methods of order r, with a Lipschitz assumption on f):

- Defining the *global error* e_n at time $t = n\Delta t$ to be

$$e_n = \|u(n\Delta t) - U_n\|$$

then an example of a convergence result is (for Runge-Kutta methods of order r, with a Lipschitz assumption on f):

- there is a $\Delta t_c > 0$ such that for any $\Delta t \in (0, \Delta t_c)$ and $n\Delta t \in [0, T]$ the global error satisfies

$$e_n \leq K\Delta t^r(e^{LT} - 1).$$

- Defining the *global error* e_n at time $t = n\Delta t$ to be

$$e_n = \|u(n\Delta t) - U_n\|$$

then an example of a convergence result is (for Runge-Kutta methods of order r, with a Lipschitz assumption on f):

- there is a $\Delta t_c > 0$ such that for any $\Delta t \in (0, \Delta t_c)$ and $n\Delta t \in [0, T]$ the global error satisfies

$$e_n \leq K\Delta t^r(e^{LT} - 1).$$

- But, we see that the bound grows exponentially in time and goes to infinity as $T \to \infty$. So what do we do if we are interested in long-time behaviour?

What else besides near term error do we need to worry about?

What else besides near term error do we need to worry about?

- Much of the time we are interested in understanding the stability properties of the invariant sets (equilibrium points, periodic solutions, chaotic attractors, *etc.*) of our model dynamical system.

What else besides near term error do we need to worry about?

- Much of the time we are interested in understanding the stability properties of the invariant sets (equilibrium points, periodic solutions, chaotic attractors, *etc.*) of our model dynamical system.
- Also, often our model contains parameters, which we may assume are constant, but may actually vary over longer time periods, or may just be subject to measurement or modeling error, so really we are dealing with a system

$$\dot{u}(t) = f(u, \mu), \quad u(0) = u_0, \quad n \in \mathbb{R}^+, \mathbb{R}, u \in \mathbb{R}^n, \mu \in \mathbb{R}^m \tag{2}$$

so we need to worry about bifurcations.

This presents us with a whole host of other discretization concerns.

This presents us with a whole host of other discretization concerns.

- How do we know whether the invariant sets and bifurcations of the discretized system reflect or correspond to the invariant sets of our original model?

- Consider a simple linear example:

$$\dot{u} = Au, \quad u(0) = U, A = \begin{pmatrix} -\mu & 0 \\ 0 & -\mu/2 \end{pmatrix}, \mu > 1.$$

The matrix A has eigenvalues $\lambda = -\mu, -\mu/2$ and so the origin is globally asymptotically stable.

- Consider a simple linear example:

$$\dot{u} = Au, \quad u(0) = U, A = \begin{pmatrix} -\mu & 0 \\ 0 & -\mu/2 \end{pmatrix}, \mu > 1.$$

The matrix A has eigenvalues $\lambda = -\mu, -\mu/2$ and so the origin is globally asymptotically stable.

- Suppose we discretize using the Euler method, then

$$U_{n+1} = (I + \Delta t A)U_n$$

- Consider a simple linear example:

$$\dot{u} = Au, \quad u(0) = U, A = \begin{pmatrix} -\mu & 0 \\ 0 & -\mu/2 \end{pmatrix}, \mu > 1.$$

  The matrix A has eigenvalues $\lambda = -\mu, -\mu/2$ and so the origin is globally asymptotically stable.

- Suppose we discretize using the Euler method, then

$$U_{n+1} = (I + \Delta t A) U_n$$

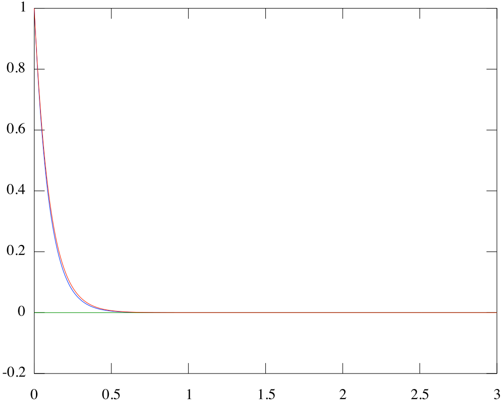
  Clearly $U = 0$ is an equilibrium point, but it is only stable provided that the modulus of the eigenvalues of $I + \Delta t A$ are less than 1, *i.e.* if $|1 - \mu \Delta t| < 1$.

So we will run into trouble when $\mu\Delta t = 2$, *i.e.* when $\Delta t = 2/\mu$.

- Suppose that $\mu = 10$ and we let $U = (1,0)^T$. Then what happens for $\Delta t = 0.1/\mu$?

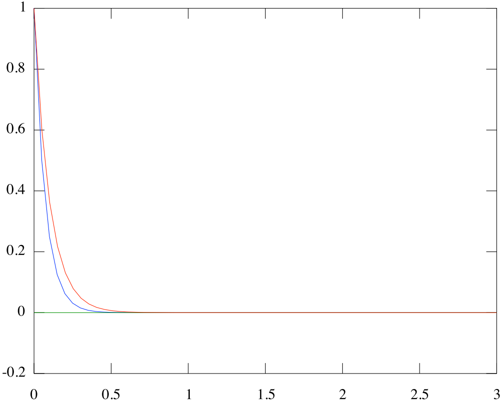
So we will run into trouble when $\mu\Delta t = 2$, *i.e.* when $\Delta t = 2/\mu$.

- Suppose that $\mu = 10$ and we let $U = (1, 0)^T$. Then what happens for $\Delta t = 0.1/\mu$?



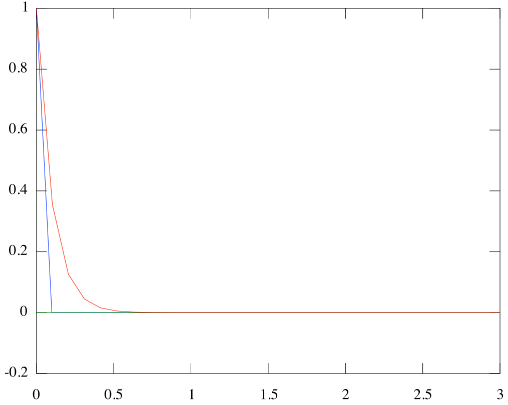
So we will run into trouble when $\mu\Delta t = 2$, *i.e.* when $\Delta t = 2/\mu$.

- Suppose that $\mu = 10$ and we let $U = (1, 0)^T$. Then what happens for $\Delta t = 0.5/\mu$?



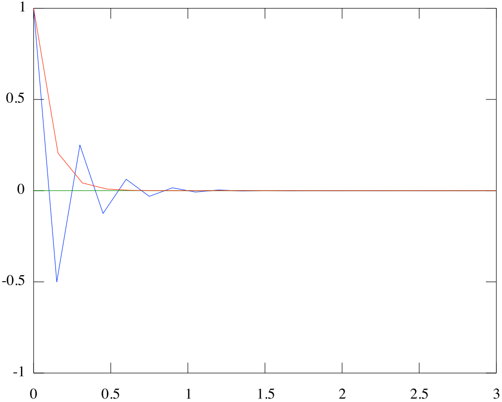
So we will run into trouble when $\mu\Delta t = 2$, *i.e.* when $\Delta t = 2/\mu$.

- Suppose that $\mu = 10$ and we let $U = (1, 0)^T$. Then what happens for $\Delta t = 1/\mu$?



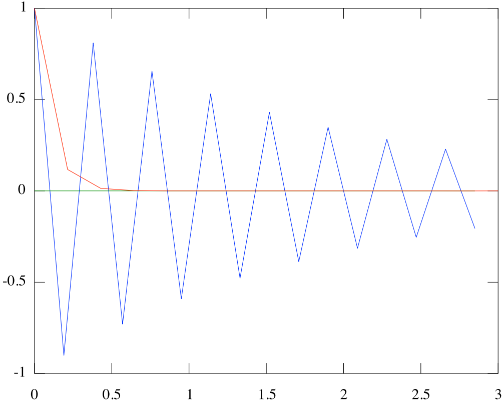
So we will run into trouble when $\mu\Delta t = 2$, *i.e.* when $\Delta t = 2/\mu$.

- Suppose that $\mu = 10$ and we let $U = (1, 0)^T$. Then what happens for $\Delta t = 1.5/\mu$?



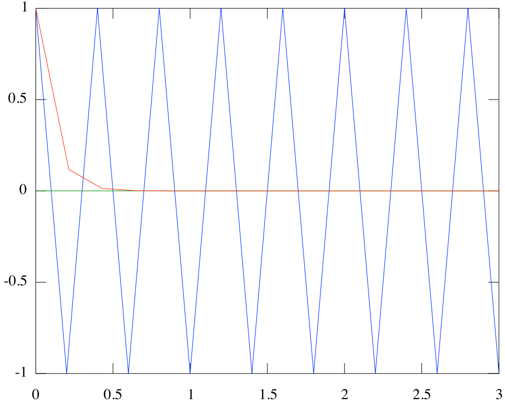
So we will run into trouble when $\mu\Delta t = 2$, *i.e.* when $\Delta t = 2/\mu$.

- Suppose that $\mu = 10$ and we let $U = (1, 0)^T$. Then what happens for $\Delta t = 1.9/\mu$?



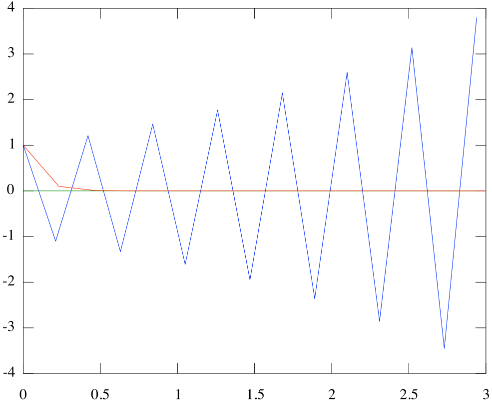
So we will run into trouble when $\mu \Delta t = 2$, *i.e.* when $\Delta t = 2/\mu$.

- Suppose that $\mu = 10$ and we let $U = (1, 0)^T$. Then what happens for $\Delta t = 2/\mu$?



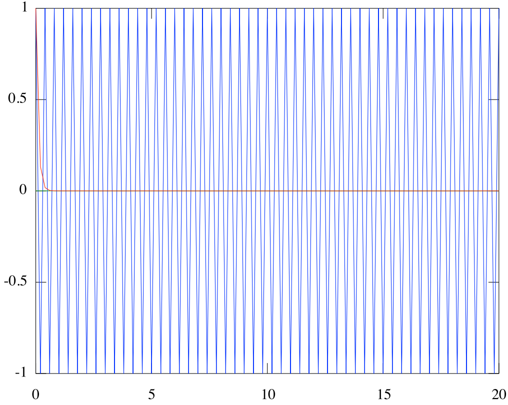
So we will run into trouble when $\mu \Delta t = 2$, *i.e.* when $\Delta t = 2/\mu$.

- Suppose that $\mu = 10$ and we let $U = (1, 0)^T$. Then what happens for $\Delta t = 2.1/\mu$?



So we will run into trouble when $\mu\Delta t = 2$, *i.e.* when $\Delta t = 2/\mu$.

- Suppose that $\mu = 10$ and we let $U = (1, 0)^T$. Then what happens for $\Delta t = 2/\mu$?



In terms of understanding the limits of numerics there are two main kinds of question that arise:

In terms of understanding the limits of numerics there are two main kinds of question that arise:

1. *Convergence*: what can be said about the large n behaviour of U_n and the large t behaviour of $u(t)$ in the limit $\Delta t \to 0$?

In terms of understanding the limits of numerics there are two main kinds of question that arise:

1. *Convergence*: what can be said about the large n behaviour of U_n and the large t behaviour of $u(t)$ in the limit $\Delta t \to 0$?
2. *Stability*: for what sort of restrictions on the time step do numerical methods replicate the large-time dynamical features of the dynamical system?

In terms of understanding the limits of numerics there are two main kinds of question that arise:

1. *Convergence*: what can be said about the large n behaviour of U_n and the large t behaviour of $u(t)$ in the limit $\Delta t \to 0$?
2. *Stability*: for what sort of restrictions on the time step do numerical methods replicate the large-time dynamical features of the dynamical system?

How do we address these?

One approach is to treat the numerical methods as discrete dynamical systems so that the theory of dynamical systems can be brought to bear on the problem.

To get a taste for what this involves we need a few more definitions. . .

A (finite-dimensional) dynamical system can be characterized as a map from vectors to vectors.

A (finite-dimensional) dynamical system can be characterized as a map from vectors to vectors.

- For a discrete dynamical system the map is

$$S^n\colon \mathbb{R}^n \to \mathbb{R}^n, \qquad U_0 \mapsto U_n = S^n U_0.$$

A (finite-dimensional) dynamical system can be characterized as a map from vectors to vectors.

- For a discrete dynamical system the map is

$$S^n\colon\ \mathbb{R}^n \to \mathbb{R}^n, \qquad U_0 \mapsto U_n = S^n U_0.$$

- For a continuous dynamical system the map is

$$S(t)\colon\ \mathbb{R}^n \to \mathbb{R}^n, \qquad u_0 \mapsto u(t) = S(t)u_0.$$

A (finite-dimensional) dynamical system can be characterized as a map from vectors to vectors.

- For a discrete dynamical system the map is

$$S^n\colon \mathbb{R}^n \to \mathbb{R}^n, \qquad U_0 \mapsto U_n = S^n U_0.$$

- For a continuous dynamical system the map is

$$S(t)\colon \mathbb{R}^n \to \mathbb{R}^n, \qquad u_0 \mapsto u(t) = S(t)u_0.$$

The maps (S^n and $S(t)$) are called *evolution semigroups*.

With the notion of a semigroup we may give a precise definition of invariant set:

With the notion of a semigroup we may give a precise definition of invariant set:

- An *invariant set* of a dynamical system is a set $E \subseteq \mathbb{R}^n$ such that $S^n E = E$ or $S(t)E = E$.

With the notion of a semigroup we may give a precise definition of invariant set:

- An *invariant set* of a dynamical system is a set $E \subseteq \mathbb{R}^n$ such that $S^n E = E$ or $S(t)E = E$.

We may see that equilibrium points and periodic solutions are invariant sets since they map onto themselves at time evolves.

- In particular, the invariant sets we are interested in are the ones that solutions (or sets of solutions) converge to as $n, t \to \infty$. These are ω-limit sets.

- In particular, the invariant sets we are interested in are the ones that solutions (or sets of solutions) converge to as $n, t \to \infty$. These are ω-limit sets.
- Thus, the kind of questions we want to ask is do the ω-limit sets of our numerical method correspond to the ω-limit sets of the dynamical system.

- A fruitful way to address this is to show that the semigroup $S(t)$ for our dynamical system ((1) or (2)) when discretized yields a semigroup $S^n_{\Delta t}$ for the numerics.

- A fruitful way to address this is to show that the semigroup $S(t)$ for our dynamical system ((1) or (2)) when discretized yields a semigroup $S^n_{\Delta t}$ for the numerics.
- When it does, then it is possible to prove results concerning the relationship between the invariant sets of $S(t)$ and $S^n_{\Delta t}$.

The kinds of things that can be proved for particular classes of numerical methods are:

The kinds of things that can be proved for particular classes of numerical methods are:

- conditions under which structural properties, *e.g.* dissipativity or conservativeness, of the vector field f are preserved.

The kinds of things that can be proved for particular classes of numerical methods are:

- conditions under which structural properties, *e.g.* dissipativity or conservativeness, of the vector field f are preserved.
- conditions for convergence of the invariant sets of $S(t)$ and $S^n_{\Delta t}$, *e.g.* equilibriua, periodic solutions, invariant manifolds, *etc.*

Other interesting things that can be proved are results concerning:

Other interesting things that can be proved are results concerning:

- the preservation of arbitrary *attracting sets* under discretization. This includes the case of chaotic attractors.

Other interesting things that can be proved are results concerning:

- the preservation of arbitrary *attracting sets* under discretization. This includes the case of chaotic attractors.
- the preservation of dynamical invariants, *e.g.* a Hamilitonian, phase volume, *etc.*, under discretization. This ensures stable numerics for computing integrable systems.

One kind of stability problem for numerical methods is the introduction of spurious fixed points, *i.e.* fixed points of $S^n_{\Delta t}$ that are not fixed points of $S(t)$.

One kind of stability problem for numerical methods is the introduction of spurious fixed points, *i.e.* fixed points of $S^n_{\Delta t}$ that are not fixed points of $S(t)$. A couple of simple results that can be proved for Runge-Kutta methods are:

One kind of stability problem for numerical methods is the introduction of spurious fixed points, *i.e.* fixed points of $S^n_{\Delta t}$ that are not fixed points of $S(t)$. A couple of simple results that can be proved for Runge-Kutta methods are:

- any fixed point of $S(t)$ is a fixed point of $S^n_{\Delta t}$;

One kind of stability problem for numerical methods is the introduction of spurious fixed points, *i.e.* fixed points of $S^n_{\Delta t}$ that are not fixed points of $S(t)$. A couple of simple results that can be proved for Runge-Kutta methods are:

- any fixed point of $S(t)$ is a fixed point of $S^n_{\Delta t}$; but
- specific conditions for critical values of Δt where spurious fixed points bifurcate from equilibria.

Take home message:

Take home message:

- it is important to be aware of the kinds of distortion that discretization can produce when using numerics to study dynamical systems;

Take home message:

- it is important to be aware of the kinds of distortion that discretization can produce when using numerics to study dynamical systems;
- treating numerics as discrete dynamical systems enables one to determine how to design numerical methods that stably compute continuous dynamical systems of various kinds.

For the project:

For the project:

- A brief survey of some of the kinds of distortion produced by discretization;

For the project:

- A brief survey of some of the kinds of distortion produced by discretization;
- Simulations to illustrate the results of theorems;

For the project:

- A brief survey of some of the kinds of distortion produced by discretization;
- Simulations to illustrate the results of theorems;
- Discretization of a model of a physical system.

Thank You!

1. Hale, J. K. (1992) "Dynamics and Numerics." Pp. 243-253 in D. S. Broomhead and A. Iserles (Ed.), *The Dynamics of Numerics and the Numerics of Dynamics*, Clarendon Press.
2. Stuart, A. M. (1994) *Numerical Analysis of Dynamical Systems*. Acta Numerica, **3**, pp. 467-572.
3. Stuart, A. M. and Humphries, A. R. (1996) *Dynamical Systems and Numerical Analysis*. Cambridge University Press.